

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	09/769,604	Confirmation No.:	4397
Applicant	:	Stephen M. Howard et al.		
Filed	:	January 25, 2001		
T.C./A.U.	:	2157		
Examiner	:	Osman, Ramy M.		
Docket No.	:	EMC-002PUS		
Customer No.	:	51576		

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sirs:

DECLARATION UNDER 37 CFR §1.131

1. I Glenn Sachar, am a co-inventor of a U.S. Patent Application entitled "Data Backup System Having a Flexible Restore Architecture," now pending before the U. S. Patent Office as Application No. 09/769,604. Stephen M. Howard, William Alton Gill, Robert and Alan Boudrie, are co-inventors and their Declarations are enclosed herewith.
2. I have reviewed the Office Action dated July 17, 2008.
3. I hereby affirm, and incorporate herein by reference, my prior Declarations under Rule 1.131 submitted in the present application.
4. Prior to December 22, 2000, the invention claimed in U.S. Patent Application No. 09/769,604 was actually reduced to practice. The claimed invention was stored in a tangible form on disk to enable EMC to provide a software release to their customers prior to December 22, 2000. The software release is EDM version 4.5.0, first released to EMC customers in March of 2000. The subject matter claimed in the '604 application,

which was reduced to practice prior to December 22, 2000, includes subject matter set forth in claim 1, for example, which requires:

A method of restoring backed up data, comprising:

retrieving, by a data backup storage system, a list of objects that are restorable by a client having a backup/restore module and a logical volume manager to communicate with a storage system, the backup storage system having a storage system interface to communicate with the storage system, a backup storage unit to store backed up data, and a network interface to communicate with the client;

displaying the list of restorable objects for browsing by a user, wherein the restorable objects are located on a plurality of physical storage devices;

generating a first list of restorable objects marked for restoration by the user, wherein each of the restorable objects is associated with a particular library, wherein the library supports at least one catalog containing information for the backed up data including media type and metadata;

submitting the first list of marked restorable objects to the backup storage system for restoration for the client;

submitting a second list of marked restorable objects to the backup storage system; and

executing, by the backup storage system, a restoration of the submitted first and second lists of marked restorable objects via a remote procedure call such that first and second restore submissions can be made prior to restore execution.

Functionality for the claimed subject matter was contained in software modules forming a part of a software release prior to December 22, 2000, as described more fully below and in the attached Exhibits.

5. Prior to December 22, 2000, the claimed invention was contained in a software release for the EDM product of EMC Corporation. Exhibit A shows a Software Reference document dated March 2000 describing the claimed invention. The claimed

subject matter is referenced at least at page 48-53 (GUI) (1-14 to 1-19), 78 (3-12), 125 (5-13), and 128-131 (5-16 to 5-20).

6. Exhibit B shows Software Release Notes document dated March 2000 the claimed subject matter is referred to at least in pages 11-12 (1-7,8).

7. During the week of September 26, 2008 and other times, the source code for the software release prior to December 22, 2000 was loaded on a machine located in Hopkinton, MA, headquarters of EMC Corporation, the assignee of the present application. The software was loaded on the machine from a version of the software created prior to December 22, 2000 stored in tangible form. The software release is EDM version 4.5.0 released in March 2000. Christopher Wroten assisted me in getting the version 4.5.0 release loaded on the machine, as affirmed in his accompanying Declaration. With version 4.5.0 loaded on the machine, a series of screen shots was generated described below demonstrating the functionality set forth in claim 1 and other claims.

8. Exhibit C contains screen shots and Exhibit D contains source code that together are used to show the claimed functionality. It is noted that the code is very well commented and routines descriptively titled to enable one having a basic familiarity with computer programming to readily understand the code. Mapping of the code to the claimed functionality is provided to expedite confirmation that the code in fact supports the claimed functionality.

9. The claimed retrieving step of claim 1 is provided in code. It is understood that no graphical user interface is contained in the code to show the step of retrieving restorable objects. The application programming interface (API) enables display of the retrieved information to the user, as shown and described below. Code block D1 includes a EDMRST_GetBackupServers function on page 3 of 36 to “allow retrieval of the EDM servers which are available to perform restores from,” a EDMRST_GetSourceHosts function on page 8-9, a GetTopLevelObjects function on pages 9-10, a

EDMRST_GetAllTopLevelObjects function on page 10, and a EDMRST_GetRestorableObjects function on page 15. These are the basic functions responsible for providing the claimed retrieving step.

10. The claimed displaying step is shown in screen shot C1 and code block D1 and D2. Screen shot C1 shows a list of restorable objects in /usr/epoch/EB/config for browsing by a user. The objects can be located on a plurality of physical storage devices, as claimed. On page 1 of 36 of code block D1, a RESTORE_API provides the recover application programming interface that serves to display the retrieved restorable objects along with the restoreRPC in code block D2 and EDMRST_GetTopLevelObjects in code block D4. A graphical user interface (GUI) or command line interface (CLI) calls EDMRST_GetTopLevelObjects which causes, via an RPC, RSTSL_GetTopLevelObjects (code block D4) to be executed in the restore process. Code Block D5 contains EDMRST_GetRestorableObjects functions to display the retrieved objects.

11. The claimed generating step is shown in screen shot C2. As can be seen, objects autoconfig.cfg, clients_installed, and defines.cfg are marked for restoration. Code block D6 contains routines EDMRST_GetMarkResults, EDMRST_MarkObject, and EDMRST_Unmarkobject. As explained on page 2 of 12, for example, the commenting in EDMRST_MarkObject describes in detail that this “function is passed a restorable object and begins to mark files for restoral based on the input criteria.” Code block D7 shows, for example, RSLmarkunm that “contains the Restore Service *Library* functions to mark and unmark objects for restoral.” Page 9 of code block D7, shows a call to mcat_getcatlm to retrieve a catalog entry.

12. The claimed step of submitting the first list of objects is shown in screen shot C2 above. Screen shot C3 shows a submit file (submit.file.4683.1) containing the marked restorable objects. The submit files are persistent. Code block D9 contains a series of functions including independent EDMRST_Submit and EDMRST_Start files. The EDMRST_start file executes the restoration of the objects contained in the submit files. On page 2 of 18, the RSTstart.c file describes that the intent of the file is “to implement

the functions that control execution of the restore for the Restore API...creation of submit objects...starting the restoral of a submit object...” On page 4, the EDMRST_Submit file describes that the “function starts the creation or update of a submit object from the currently marked restorable objects.”

13. The claimed step of submitting a second list of marked restorable objects is shown in screen shot C4 in which files are marked and submitted. D9 described above provides the submit functionality.

14. The claimed executing step is shown in screen shot C5 showing a successful restoration. Screen shot C6 shows the existence of the persistent submit files. The persistent submit files are independent of the restore execution to enable “executing, by the backup storage system, a restoration of the submitted first and second lists of marked restorable objects via a remote procedure call such that first and second restore submissions can be made prior to restore execution.” Code block D9 described above provides the execution functionality for the restore.

15. Screen shots C7, along with screen shots C2 and C4, each generate one of the submit files shown in C6. By providing independent submit and restore code functions, as described above, any number of lists of marked objects can be submitted prior to execution.

16. Code block D8 contains a series of API functions to retrieve information about the media needed for restoration. A function executed via a RPC call in the restore process retrieves information about the media needed for the restore.

17. Code block D10 shows code providing the functionality of claim 2, which requires “executing the first and second lists of marked restorable objects concurrently.” Concurrent execution of first and second lists of marked restorable objects is provided via multiple restore engine execution controlled by the Dispatch Daemon and via the user

interface making multiple calls to EDMRSTstart after multiple calls to EDMRSTsubmit in code block D9 described above.

18. Code block D10 shows code providing the functionality of claim 3, which requires “initiating a restore session for the client.” In particular, EDMRST_Initialize is a user call to initiate the restore session.

19. Code block D10 shows the code providing the functionality of claim 4, which requires “creating a restore engine process for the retrieving, browsing, submitting, and executing of restore objects.” EDMRST_Initialize uses RPC calls in dispatch_daemon to call Dispatch Daemon code contained in EDMDDispatch.c, EDMDDispatchService.c, and EDMDDispatchSession.c.

20. Code block D11 shows the code providing the functionality of claim 8, which requires a “method according to claim 4, wherein the restore engine process runs on a backup data storage server and further including creating a work item restore process on the backup data server, a server restore process for generating a stream of data to be restored, and a client restore process for receiving the data stream.” The below shows an exemplary representation:

RSTS_L_Start

execute WorkItemRestore

 RunWorkItemRestore

 RunWorkItemRestore for trail

 InitiateWorkItemRestore

 Startup AuxProcess (creates the triangle)

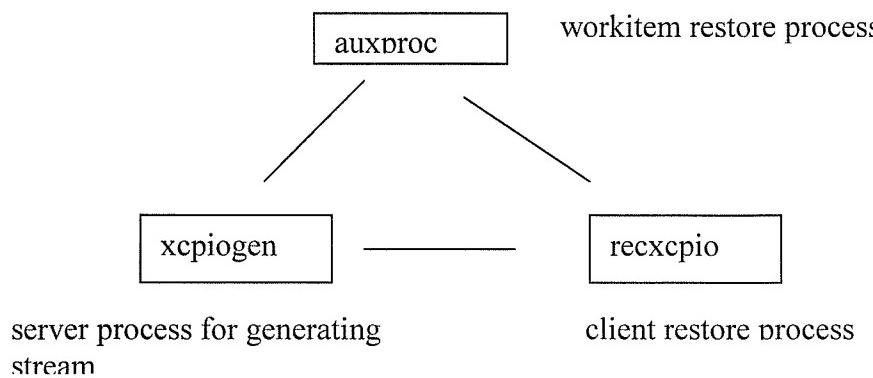
 StartWorkItemRestore

 -sends client restore process info to main in RSLauxmain.c

 -forks xcpioen:server process for generating stream

 -calls z_fcmdfilter to start client restore process

 Triangle



21. Code block D12 shows the code providing the functionality of claim 9, which requires a “method according to claim 4, further including detecting and identifying libraries that support associated catalogs of backed up data for processing of backed up data by the restore engine process” in RSTSL_Initialize function invoked at the start of the restore session. Init_plugins is called to find and detect libraries. Claims 10, 11 14, 17, and 21 are also shown in code block D12.

22. The source code providing the claimed subject matter existed in a tangible form prior to December 22, 2000. The software existed in a tangible form stored so as to be able to provide EMC customers with Release Version 4.5.0 in March of 2000. The source code existed on disk including an article comprising a computer readable medium containing instructions that when executed by a machine provide the functionality set forth in claim 1 and other claims.

23. All of the acts referred to herein took place in the United States.

24. All of the statements made herein are of my own knowledge and are true, these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment or both, under § 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application and any patent issuing thereon, or any patent to which this verified statement is applied.

12/08/08
Date of Signature

By: Glenn R. Sachar
Glenn R. Sachar

emc-002 sachar dec.doc